

### **Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

### **Listing of Claims:**

1. (Currently Amended) A method in a data processing system for processing a ~~Java-server~~ JAVASERVER page, the method comprising:  
translating the ~~Java-server~~ JAVASERVER page into a document object model object;  
configuring a set of visitor classes for invocation in a selected sequence; [[and]]  
processing the document object model using the set of visitor classes in the selected sequence to perform a desired set of custom functions on the document object model; and  
storing results, as processing the document object model object occurs, by selected method in the methods, in a hash map, wherein the results in the hash map are used by subsequently invoked methods, wherein the subsequently invoked method does not convert the results into a second document object model object.
2. (Currently Amended) The method of claim 1 further comprising:  
validating syntax in the ~~Java-server~~ JAVASERVER page.
3. (Original) The method of claim 1, wherein the set of visitor classes for invocation in the selected sequence is defined in a configuration file.
4. (Original) The method of claim 3, wherein the configuration file is an extensible markup language file.
5. (Original) The method of claim 3, wherein the selected sequence is defined in the configuration file.
6. (Original) The method of claim 1, wherein  
the document object model object includes a set of nodes and wherein the processing step includes:

invoking methods in the set of visitor classes on each node in the set of nodes in the selected sequence.

7. (Canceled)

8. (Currently Amended) The method of claim 1, wherein the ~~Java-server~~ JAVASERVER page is translated into a document object model object using a document object model generator.

9. (Currently Amended) The method of claim 2, wherein the ~~Java-server~~ JAVASERVER page is validated using a ~~Java-server~~ JAVASERVER page translator.

10. (Currently Amended) The method of claim 9, wherein the ~~Java-server~~ JAVASERVER page translator invokes a visitor class to validate elements in the document object model object against a syntax for a ~~Java-server~~ JAVASERVER page specification.

11. (Currently Amended) The method of claim 1 wherein results from processing [[by]] using a first visitor class in the set of visitor classes are passed to a second visitor class in the set of visitor classes.

12. (Currently Amended) A data processing system for processing a ~~Java-server~~ JAVASERVER page, the data processing system comprising:  
a memory containing a set of instructions; and  
a processing unit, responsive to execution of the set of instructions, translating means for translating the ~~Java-server~~ JAVASERVER page into a document object model object; configuring means for configuring a set of visitor classes for invocation in a selected sequence; and processing means for processing the document object model using the set of visitor classes in the selected sequence to perform a desired set of custom functions on the document object model; and for storing results, as processing the document object model object occurs by selected method in the methods, in a hash map, wherein the results in the hash map are used by subsequently invoked methods.

13. (Currently Amended) The data processing system of claim 12 wherein the processing unit, responsive to execution of the set of instructions, further comprises further comprising: validating means for validating syntax in the ~~Java-server~~ JAVASERVER page.

14. (Original) The data processing system of claim 12, wherein the set of visitor classes for invocation in the selected sequence is defined in a configuration file.
15. (Original) The data processing system of claim 14, wherein the configuration file is an extensible markup language file.
16. (Original) The data processing system of claim 14, wherein the selected sequence is defined in the configuration file.
17. (Currently Amended) The data processing system of claim 12, wherein the document object model object includes a set of nodes and wherein the processing ~~[[means]]~~ includes:  
~~means for~~ invoking methods in the set of visitor classes on each node in the set of nodes in the selected sequence.
18. (Canceled)
19. (Currently Amended) The data processing system of claim 12, wherein the ~~Java-server~~ JAVASERVER page is translated into a document object model object using a document object model generator.
20. (Currently Amended) The data processing system of claim 13, wherein the ~~Java-server~~ JAVASERVER page is validated using a ~~Java-server~~ JAVASERVER page translator.
21. (Currently Amended) The data processing system of claim 20, wherein the ~~Java-server~~ JAVASERVER page translator invokes a visitor class to validate elements in the document object model object against a syntax for a ~~Java-server~~ JAVASERVER page specification.
22. (Original) The data processing system of claim 1, wherein results from processing by a first visitor class in the set of visitor classes are passed to a second visitor class in the set of visitor classes.

23. (Currently Amended) A computer program product in ~~computer-readable recordable-~~  
~~type~~ medium for processing a ~~Java-server~~ JAVASERVER page, the computer program product  
comprising:

first instructions for translating the ~~Java-server~~ JAVASERVER page into a document  
object model object;

second instructions for configuring a set of visitor classes for invocation in a selected  
sequence; [[and]]

third instructions for processing the document object model using the set of visitor  
classes in the selected sequence to perform a desired set of custom functions on the document  
object model; and

fourth instructions for storing results, as processing the document object model object  
occurs by selected method in the methods, in a hash map, wherein the results in the hash map are  
used by subsequently invoked methods.

24. (Currently Amended) The computer program product of claim 23 further comprising:  
fourth instructions for validating syntax in the ~~Java-server~~ JAVASERVER page.

25. (Original) The computer program product of claim 23, wherein the set of visitor classes  
for invocation in the selected sequence is defined in a configuration file.

26. (Original) The computer program product of claim 25, wherein the configuration file is  
an extensible markup language file.

27. (Original) The computer program product of claim 25, wherein the selected sequence is  
defined in the configuration file.

28. (Original) The computer program product of claim 23, wherein the document object  
model object includes a set of nodes and wherein the third instructions includes:

sub instructions for invoking methods in the set of visitor classes on each node in the set  
of nodes in the selected sequence.

29. (Canceled)

30. (Currently Amended) The computer program product of claim 23, wherein the ~~Java server~~ JAVASERVER page is translated into a document object model object using a document object model generator.
31. (Currently Amended) The computer program product of claim 24, wherein the ~~Java server~~ JAVASERVER page is validated using a ~~Java-server~~ JAVASERVER page translator.
32. (Currently Amended) The computer program product of claim 31, wherein the ~~Java server~~ JAVASERVER page translator invokes a visitor class to validate elements in the document object model object against a syntax for a ~~Java-server~~ JAVASERVER page specification.
33. (Original) The computer program product of claim 23, wherein results from processing by a first visitor class in the set of visitor classes are passed to a second visitor class in the set of visitor classes.
34. (Currently Amended) A data processing system for processing a ~~Java-server~~ JAVASERVER page, the data processing system comprising:  
a bus system;  
a memory connected to the bus system, wherein the memory includes a set of instructions;  
a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to translate the ~~Java-server~~ JAVASERVER page into a document object model object; configure a set of visitor classes for invocation in a selected sequence; [[and]] process the document object model using the set of visitor classes in the selected sequence to perform a desired set of custom functions on the document object model; and storing results, as processing the document object model object occurs by selected method in the methods, in a hash map, wherein the results in the hash map are used by subsequently invoked methods.